НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«На правах рукопису»
УДК _____

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“___” _____ 2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 113 «Прикладна математика» _____
(код і назва)

на тему: Протоколи узгодження в розподілених системах_____
_____
_____

Виконав (-ла): студент (-ка) __2__ курсу, групи ФІ-73 мп
(шифр групи)

Карпець Антон Сергійович _____ _____
(прізвище, ім’я, по батькові) (підпис)

Керівник професор, д.т.н, Кудін А. М. _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____ _____ _____
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент_____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти: другий (магістерський) за освітньо–професійною програмою

Спеціальність: 113 «Прикладна математика»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
«___»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Карпцю Антону Сергійовичу _____
(прізвище, ім'я, по батькові)

1. Тема дисертації Протоколи узгодження в розподілених системах_____
_____
_____ ,
науковий керівник дисертації Кудін Антон Михайлович, професор, д.т.н ___ ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від _____р. № _____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження _____
_____

4. Предмет дослідження (Вхідні дані – для магістерської дисертації за освітньо–професійною програмою)
_____
_____
_____

5. Перелік завдань, які потрібно розробити _____
_____
_____
_____
_____
_____

6. Орієнтовний перелік ілюстративного матеріалу _____
_____
_____

7. Орієнтовний перелік публікацій _____
_____
_____

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка

Студент _____ _____
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ _____
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Роботу виконано на 57 аркушах, вона містить перелік посилань на використані джерела з 13 найменувань. В роботі наведено 1 рисунок.

Мета даної дипломної роботи полягає в тому, щоб запропонувати нові підходи до розробки протоколів узгодження в розподілених криптовалютних системах, що дозволить знизити ресурсоємність даного процесу, а також розширити коло учасників консенсусу.

Об'єктом дослідження є процес досягнення узгодження в розподілених криптовалютних системах.

Предметом дослідження є застосування криптографічних схем розділення секрету до існуючих протоколів консенсусу.

У роботі запропоновано модифікації існуючих протоколів консенсусу, які дозволяють розширити коло учасників даного процесу шляхом розподілення додаткової інформації та підвищити швидкість росту економіки окремої криптовалютної системи, а також наведені аналітичні оцінки деяких протоколів та проведено порівняльний аналіз із існуючими рішеннями.

ПРОТОКОЛИ КОНСЕНСУСУ, БЛОКЧЕЙН

РЕФЕРАТ

Дипломная работа выполнена на 57 листах, она содержит список ссылок на использованные источники с 13 наименований. В работе приведен 1 рисунок.

Цель данной дипломной работы состоит в том, чтобы предложить новые подходы к разработке протоколов согласования в распределенных криптовалютных системах, что позволит снизить ресурсозатратность данного процесса, а также расширить круг участников консенсуса.

Объектом исследования является процесс достижения согласования в распределенных криптовалютных системах.

Предметом исследования является применение криптографических схем распределения секрета к существующим протоколам консенсуса.

В работе предложено модификации существующих протоколов согласования, которые позволят расширить круг участников данного процесса путем распределения дополнительной информации и повысить скорость роста экономики отдельной криптовалютной системы, а также приведены аналитические оценки некоторых протоколов и проведено сравнительный анализ с существующими решениями.

ПРОТОКОЛЫ КОНСЕНСУСА, БЛОКЧЕЙН

ABSTRACT

The thesis is presented in 57 pages. It contains bibliography of 13 references. 1 figure is given in the thesis.

The object is a process of consensus achievement in distributed cryptocurrency systems.

The subject is application of cryptographic secret sharing schemes to the existing consensus protocols.

This thesis presents the new modifications of an existing consensus protocols, which allows to increase number of users of this process and also improve economy state of cryptocurrency system.

CONSENCUS PROTOCOLS, BLOCKCHAIN

ЗМІСТ

Перелік умовних позначень, скорочень і термінів 7
Вступ 8
1 Концепція розподілених криптовалютних систем 10
1.1 Опис технології блокчейн 11
1.2 Протоколи консенсусу в розподілених системах 13
1.2.1 Протокол консенсусу Proof of Work 17
1.2.2 Протокол консенсусу Proof of Stake 18
1.2.3 Delegated Proof of Stake 19
1.2.4 Протокол консенсусу Proof of Activity 21
1.2.5 Протокол консенсусу Proof of Burn 22
1.2.6 Протокол консенсусу Proof of Importance 23
1.3 Протоколи розділення секрету 27
1.3.1 Схема Блеклі 29
1.3.2 Схема Шаміра 30
1.3.3 Схема Асмута - Блума 32
1.3.4 Схеми розподілу з перевіркою секрету 32
Висновки до розділу 1 35
2 Побудова модифікацій протоколів консенсусу 36
2.1 Лотерея 37
2.2 Використання IP адрес у якості цінного ресурсу 41
Висновки до розділу 2 45
3 Аналіз протоколів консенсусу 46
3.1 Proof of Work 46
3.2 Proof of Stake 48
3.3 Proof of Importance 50
Висновки до розділу 3 52
Висновки 54
Перелік посилань 56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

PoW — (англ. «Proof of Work») доведення виконання роботи

PoS — (англ. «Proof of Stake») доведення володіння часткою

UTXO — (англ. «Unspent Transaction Output») сукупність виходів невикористаних транзакцій

CAP — (англ. «Consistency, Availability, Partition tolerance») узгодженість, доступність, стійкість до розділення

DPoS — (англ. «Delegated Proof of Stake») делеговане доведення володіння часткою

PoA — (англ. «Proof of Activity») доказ діяльності

NEM — New Econmy Movement

ВСТУП

Актуальність дослідження.

Нині все більше повсякденних завдань покладається на інформаційні технології. Поступово, вони з реального світу переходять в простір Інтернет. Зараз стало можливим спілкуватися, робити покупки, керувати своїми грошовими ресурсами за допомогою комп'ютера чи смартфона. Відповідно, було також запропоновано і способи утримання та обміну коштами в Інтернеті. У якості одного з таких підходів було запропоновано криптовалюти. Для забезпечення їх працездатності необхідна наявність певного кола людей, здатних виконувати певну роботу. Деякі з існуючих рішень не є ефективними та не дають можливості долучатися до них всім бажаючим. Тому важливим є впровадження механізмів, здатних підтримувати та стимулювати таких користувачів для подальшого розвитку цих систем.

Мета дослідження полягає в тому, щоб запропонувати нові підходи до розробки протоколів узгодження в розподілених криптовалютних системах, що дозволить знизити ресурсоємність даного процесу, а також розширити коло учасників консенсусу. Для досягнення даної мети необхідно розв'язати **задачу дослідження**, яка полягає у відшуканні способів модифікації протоколів. Для розв'язання задачі необхідно вирішити такі завдання:

1) провести огляд існуючих протоколів узгодження та виявити їх недоліки;

2) розглянути та обрати спосіб розповсюдження додаткової інформації в системі з метою надання переваги більшому колу користувачів;

3) обрати спосіб впровадження даного механізму в існуючий протокол;

4) провести аналіз отриманого підходу.

Об'єктом дослідження є процес досягнення узгодження в розподілених криптовалютних системах.

Предметом дослідження є застосування схеми розділення секрету до існуючих протоколів консенсусу.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи теорії імовірностей, математичного аналізу та абстрактної алгебри.

Наукова новизна отриманих результатів полягає у тому, що вперше у протоколах консенсусу криптовалютних систем було застосовано схему розподілу секрету.

Практичне значення результатів полягає у тому, що було запропоновано внесення додаткових процесів в схему досягнення узгодження для того, щоб періодично урівнювати можливості користувачів з різним рівнем цінного ресурсу.

1 КОНЦЕПЦІЯ РОЗПОДІЛЕНИХ КРИПТОВАЛЮТНИХ СИСТЕМ

Нині торгівля в Інтернеті покладається практично повністю на фінансові установи, що використовуються у якості довірених третіх сторін для виконання обов'язків щодо обробки електронних платежів. Хоча система працює досить добре для переважної більшості операцій, в ній все-таки присутні слабкі місця, які виникають в наслідок використання моделі довіри. Повністю незворотні транзакції дійсно неможливі при такому підході, оскільки фінансові установи не можуть уникати посередницьких суперечок. В результаті наявності посередництва збільшуються операційні витрати, що призводить до обмеження мінімального розміру транзакції та робить недоцільним проведення повсякденних транзакцій невеликого розміру. Окрім того, ціна збільшується також і за рахунок неможливості проведення безповоротних платежів. Можливість повернення транзакції вимагає наявності довірених посередників. Продавці в Інтернет-системах мають бути обережними зі своїми покупцями, в наслідок чого від них вимагається більше інформації, ніж за наявності безповоротних транзакцій. Таких ускладнень можливо уникнути лише за використання звичайних грошей, проте не існує механізму проведення електронних платежів без довіреної особи.

Таким чином, виникає необхідність у системі електронних платежів, що опирається на криптографічну стійкість замість механізмів довіри, оскільки це дозволить будь-яким двом сторонам проводити транзакції безпосередньо – без використання посередників. Транзакції, які практично неможливо скасувати, захищають продавців від зловмисників, а для захисту покупців можна організувати механізм групового підтвердження [1].

У зв'язку з цим була запропонована перша криптовалютна система,

що базується на технології «блокчейн». В даному розділі буде описано принципи роботи таких систем, механізми, що використовуються в них для забезпечення захисту від зловмисних дій, а також розглянуто їх переваги та недоліки.

1.1 Опис технології блокчейн

Блокчейн, тобто ланцюг блоків транзакцій (англ. «Blockchain», від «block» – блок, «chain» – ланцюг) – розподілена база даних, яка підтримує перелік записів, так званих блоків, що постійно зростає. База захищена від підробки та переробки. Кожен блок містить часову мітку та посилання на попередній блок.

Першочергово, для побудови блокчейну необхідно обрати криптографічно стійку геш-функцію, що володіє певним набором властивостей. Тому введемо поняття геш-функції та опишемо необхідні властивості, що забезпечують коректну роботу технології блокчейн.

Стійкою до колізій важкооборотною геш-функцією називається відображення виду:

$$h(x) : \{0,1\}^{l(n)} \to \{0,1\}^n,$$

що переводить бітовий рядок довільної довжини у бітовий рядок фіксованої довжини n та має наступні властивості:

– ефективно обчислюється;

– стійка до пошуку прообразів. Нехай $M : \{0,1\}^n \to \{0,1\}^{l(n)}$ – супротивник, який працює за час T. Тоді:

$$Pr_{x,M}\{h(M(h(x))) = h(x)\} \leq \epsilon;$$

– стійка до пошуку другого прообразу; Нехай $M' : \{0,1\}^{l(n)} \to \{0,1\}^{l(n)}, \forall x M`(x) \neq x$ – супротивник, який працює за час

T. Тоді:

$$Pr_{x,M'}h(M'(h(x))) = h(x) \leq \epsilon;$$

– стійка до пошуку колізій. Нехай $F : K \to \{0,1\}^{l(n)} \times \{0,1\}^{l(n)} \cup \{0\}$ – супротивник, який працює за час $\leq T$, K - множина параметрів, та повертає пару (x, x') таку, що $h_k(x) = h_k(x')$ або повертає 0. Тоді:

$$Pr_{k,F}\{F(k) \neq 0\}.$$

Виконання даних властивостей є важливим при побудові блокчейну [2].

Побудова блокчейну нероздільно пов'язана з використанням геш-вказівників. Геш-вказівники – це структури даних, які містять в собі вказівник на місце збереження деяких даних разом з гешом цих даних. Таким чином, геш-вказівник дозволяє не просто отримати інформацію за її адресою, а і впевнитися в її незмінності з часу його створення.

Визначивши поняття геш-вказівника, визначимо блокчейн, як структуру даних, що являє собою зв'язний список, для побудови якого замість звичайних вказівників використовуються геш-вказівники. Таким чином, утворюється ланцюжок блоків даних, кожен з яких додатково містить в собі значення вказівника та значення гешу попереднього блоку. Перший блок містить в собі геш так званого генезис-блоку (genesis block), вміст якого визначається наперед. При цьому, користувач блокчейну зберігає геш-вказівник на останній блок в місці, недоступному іншим користувачам.

Блокчейн дозволяє утворити журнал цифрового пломбування (англ. – «tamper-evident log»), тобто такий журнал, підроблення записів в якому неможливо приховати. Дана властивість досягається завдяки використанню геш-вказівників.

В цьому випадку метою зловмисника буде здійснення підміни даних, що містяться в одному з блоків всередині ланцюга. Для цього він після підміни даних в блоці k повинен буде також підмінити значення гешу в блоці $k+1$. Аналогічним чином для уникнення неспівпадінь між даними в

блоках та їх гешами потрібно провести підміну у всіх блоках, починаючи з k-го і закінчуючи останнім. Зловмисник може спробувати провести підміну даних та гешів у всіх блоках, які слідують після блоку k, проте такий підхід є хибним, оскільки користувач зберігає у себе значення гешу останнього блоку, де його складно підмінити. В такому разі, практично неможливо внести зміни непомітно для користувача. Тому, геш-вказівник на останній блок даних фактично є індикатором того, чи була проведена підміна.

Варто зазначити, що для утворення геш-вказівника, як зазначалося вище, використовується стійка до колізій геш-функція, тому практично виключається можливість здійснення підміни даних таким чином, щоб значення гешу залишалось незмінним [3].

Утворений блокчейн не має конкретного місця збереження, його копія зберігається на кожному вузлі, що має до нього доступ та приймає участь у його формуванні.

1.2 Протоколи консенсусу в розподілених системах

Консенсус – загальна згода при вирішенні конфліктів у прийнятті рішень, яка характеризується відсутністю принципових заперечень у більшості зацікавлених осіб; рішення на основі загальної згоди без проведення голосування, якщо проти нього ніхто не заперечує або при виключенні думок деяких (меншості) незгодних з рішенням учасників. Також під даним терміном мається на увазі процес знаходження, пошуку або вироблення рішення, яке б задовольнило всіх учасників. У вузькому сенсі, який застосовується у криптографії, консенсус (протокол консенсусу) є саме процедурою прийняття рішення. Його мета – забезпечити узгодженість поточного стану мережі між учасниками після додавання до неї нової інформації. Таким чином, протокол консенсусу

гарантує, що ланцюг транзакцій в мережі є вірним.

Криптовалютна система, така як Bitcoin, містить інформацію про стан, що дозволяє користувачам отримувати значення їх балансів. Для Bitcoin стан системи являє собою сукупність виходів невикористаних транзакцій (UTXO), де кожен вихід криптографічно заблокований, вимагаючи від користувача пред'явлення підтвердження права власності на витрату UTXO. Замість того, щоб явним чином зберігати баланс, Bitcoin та інші цифрові валюти підтримують повну історію транзакцій користувачів, яка може бути використана для виведення поточного стану та всіх минулих станів системи. Транзакція представляє собою деякі атомарні зміни до стану системи. Транзакції згруповані в блоки; блоки утворюють упорядкований ланцюг під назвою блокчейн. Щоб організувати блокчейн, кожен блок містить посилання на попередній блок. Перший блок в ланцюзі – блок genesis – не має попереднього блоку і, як правило, його значення задається в протоколі деякою константою. Нові блоки формуються відповідно до набору правил, встановлених протоколом криптовалюти. Важливою функцією цих правил є захист від атак на блокчейн та досягнення консенсусу у випадку появи декількох альтернативних версій блокчейн. Протоколи Proof of Work та Proof of Stake стосуються двох видів обмежень на дійсні блоки і передбачають два різні механізми консенсусу.

Визначення 1. Користувач працює над блоком B, якщо він намагається знайти блок, на який посилається блок B як попередній блок.

Криптовалютна система може розглядатися як розподілена база даних, копії якої має кожен учасник інфраструктури для обміну інформацією через peer-to-peer протокол. З точки зору CAP теореми, криптовалютні системи мають властивість доступності (availability) – кожен запит отримує відповідь та здатні до розділення (partition-tolerance) – система буде працювати при умові, що деякі її вузли (nodes) відключаються. В деякі проміжки часу різні користувачі

системи будуть спостерігати різні стани системи як поточні. Така невідповідність виникає, коли новий геш блоку був отриманий, але ще не був переданий всім користувачам системи. Щоб отримати узгодженість (eventual consistency), в протоколі консенсусу повинна виконуватися наступна вимога:

Умова 1. Користувач, який створив наступний блок, повинен поширити його в мережі негайно.

В іншому випадку система буде неузгодженою, що може призводити до розділення на декілька гілок, що зображено на рис. 1.1.

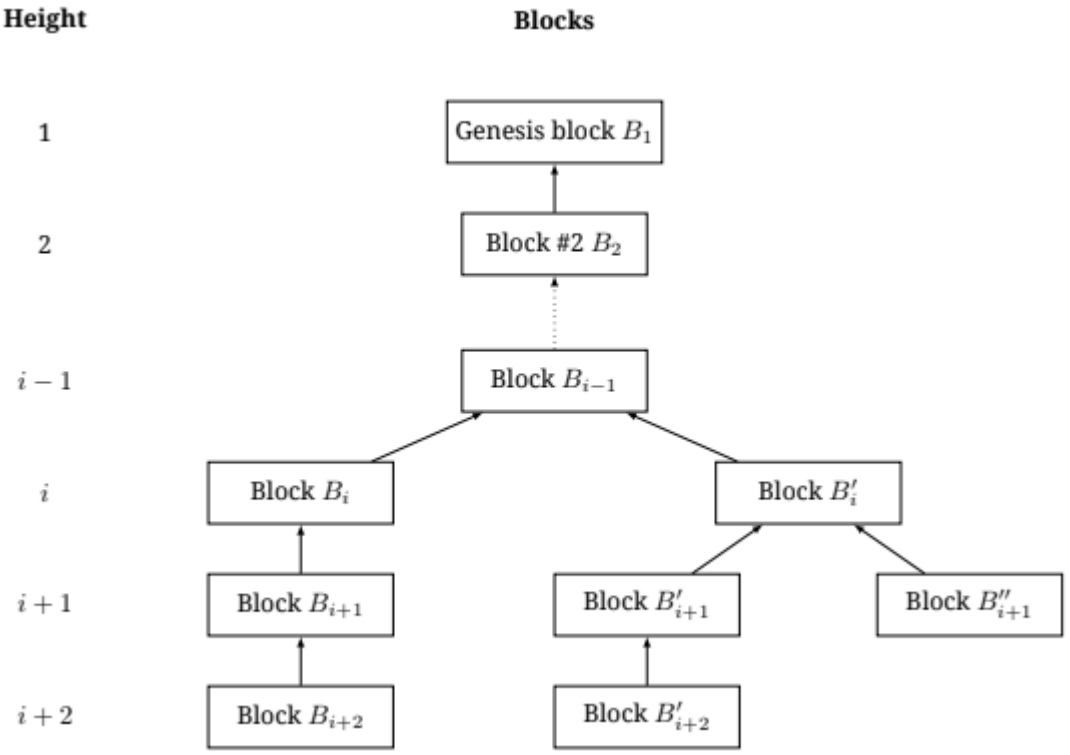

Рис. 1.1: Розгалуження блокчейну

Для того, щоб перешкоджати навмисному розгалуженню, в протоколі консенсусу потрібно врахувати наступну вимогу:

Умова 2. Користувачів необхідно обмежувати від створення блоків, що посилаються на проміжні ланки блокчейну. Точніше, якщо існує відомий блок B', який посилається на блок B, користувач не повинен мати можливість формувати блок на основі блоку B.

У ситуації, зображеній на рис. 1.1, лише три блоки можуть бути використані у якості бази для створення нового блоку, згідно з умовами 1 і 2: B_{i+2}, B''_{i+1} та B'_{i+2}. Існують додаткові обмеження на дійсний блокчейн; в більшості випадків ці умови становлять вибір ланцюга з максимальною кількістю блоків (що виключає B''_{i+1} з переліку блоків, на основі яких користувач міг би побудувати наступний блок). Метою правил консенсусу є забезпечення вибору єдиного ланцюга всіма користувачами; однак, іноді цей вибір залежить від користувача. Наприклад, якщо є кілька Bitcoin ланцюгів з однаковою довжиною, користувачі повинні обирати той, який вони отримали першим. Таким чином, неможливо застосувати такий підхід до вибору блоку для всіх користувачів.

Для того, щоб система залишалася узгодженою, її протокол консенсусу повинен також задовольняти наступній вимозі:

Умова 3. Правила консенсусу повинні бути побудовані таким чином, щоб розв'язувати проблему вибору гілки у блокчейні, тобто одна з конкуруючих гілок повинна стати головною за розумну кількість часу.

У даній роботі буде використано наступні два терміни для позначення формування нового блоку при двох різних протоколах консенсусу:

Визначення 2. Процес отримання нового дійсного блоку з використанням протоколу Proof of Work називається майнінг (англ. «mining» – видобування).

Визначення 3. Процес отримання нового дійсного блоку з використанням протоколу Proof of Stake називається мінтінг (англ. «minting» – чеканка) [4].

1.2.1 Протокол консенсусу Proof of Work

Розглянемо Bitcoin як приклад криптовалютної системи, захищеної алгоритмом Proof of Work. Кожен блок в Bitcoin складається з двох частин:

– блок-заголовок, який містить ключові параметри, включаючи час створення блоку, посилання на попередній блок і корінь дерева Меркле блоку транзакцій;

– блок-список транзакцій.

У якості посилання на конкретний блок буде використано його двічі загешований за допомогою функції $SHA-256$ заголовок, в результаті чого отримується ціле число з інтервалу $[0, 2^{256}-1]$. Для врахування різних можливих реалізацій, будемо використовувати загальну хеш-функцію $hash()$ зі змінним числом аргументів і областю значень $[0, M]$.

Посилання на блок використовується у протоколі Proof of Work. Для того, щоб блок вважався дійсним, його посилання повинно не перевищувати деякий поріг:

$$hash(B) \leq M/D;$$

де $D \in [1, M]$ – цільова складність. Немає відомого іншого способу знайти значення B, яке задовольняє умові, описаній вище, крім як перебір всіх можливих значень ітеративно. Чим вище D, тим більше ітерацій необхідно для знаходження дійсного блоку. Кількість операцій, необхідних для знаходження дійсного блоку в середньому, становить D.

Основним недоліком даного протоколу вважається його висока ресурсозатратність з точки зору обчислювальних потужностей, що практично унеможливлює майнінг для користувачів без відповідного дорогого апаратного забезпечення, а також з точки зору електроенергії [4].

1.2.2 Протокол консенсусу Proof of Stake

Proof of Stake – це протокол консенсусу для криптовалют, що є альтернативою Proof of Work, який використовується в Bitcoin. Основними задекларованими перевагами Proof of Stake є відсутність ресурсозатратних обчислень і, отже, нижчий вхідний бар'єр для винагородження за генерування нових блоків.

У алгоритмах Proof of Stake нерівість

$$hash(B) \leq M/D;$$

змінюється в залежності від балансу користувача конкретного криптовалютного PoS протоколу, а не від властивостей блоку. Розглянемо користувача з адресою A і балансом $bal()$. Зазвичай в алгоритмах Proof of Stake використовується наступна умова:

$$hash(hash(B_{prev}, A, t)) \leq bal(A) \cdot M/D;$$

де B_{prev} – блок, на основі якого будується наступний блок, t – мітка часу.

На відміну від попереднього випадку, єдина змінна, яку користувач може змінити – це мітка часу t. Баланс за адресою заблоковано протоколом; наприклад, протокол може розраховувати баланс на основі суми коштів, які не змінювалися протягом дня. Крім того, криптосистема з алгоритмом PoS може використовувати сукупність виходів невикористаних транзакцій (UTXO), як це зроблено в Bitcoin - в цьому випадку баланс дійсно заблокований. Протокол Proof of Stake вводить обмеження на можливі значення t. Наприклад, якщо t не повинно відрізнятися від часу UTC на вузлах мережі більш ніж на годину, тоді користувач може використати не більше 7200 значень t. Таким чином, в даному протоколі немає ресурсозатратних розрахунків.

Разом із адресою та міткою часу t, що задовольняє нерівності вище,

користувач повинен надати підтвердження права власності пред'явленої адреси. Для цього користувач може підписати нещодавно отриманий блок своїм підписом. Для отримання дійсного підпису потрібно мати таємний ключ, що відповідає адресі A.

Час пошуку блоку для адреси експоненціально розподіляється з параметром $bal(A)/D$. Отже, реалізація Proof of Stake є справедливою: ймовірність згенерувати дійсний блок дорівнює співвідношенню залишку коштів користувача на загальну суму коштів в обігу. Час для пошуку блоку для всієї мережі розподіляється експоненціально з параметром $\sum_a bal(a)/D$.

Таким чином, якщо грошова маса $\sum_a bal(a)/D$ фіксована або зростає з передбачуваною швидкістю, складність D повинна бути відома заздалегідь:

$$D = \frac{1}{T_{ex}} \sum_a bal(a).$$

де T_{ex} позначає позначаючи очікуваний час між блоками. На практиці, D необхідно коригувати на основі останніх блоків, оскільки не всі мінтери беруть участь у мінтингу [4].

1.2.3 Delegated Proof of Stake

Delegated Proof of Stake (DPoS) – це загальний термін, що описує розвиток основного протоколу консенсусу PoS. До DPoS належать такі протоколи, як BitShares, Slasher і Tendermint. У цих протоколах блоки добуваються заздалегідь заданим набором користувачів системи (делегатами), які отримують винагороду за виконання свого обов'язку і несуть покарання за зловмисну поведінку. У алгоритмах DPoS делегати беруть участь у двох процесах:

– створення блоку транзакцій;

– перевірка дійсності сформованого блоку шляхом цифрового підписання.

Поки блок створюється одним користувачем, для того, щоб вважатися дійсним, він, як правило, повинен бути підписаним більше, ніж одним делегатом.

Список користувачів, які мають право на підпис блоків, періодично змінюється з використанням певних правил; наприклад, у протоколі Slasher делегати для кожного блоку вибираються на основі їх частки та історії блокчейну. Кількість делегатів для кожного блоку зазвичай мала; винятком є Tendermint, для якого кожен блок може бути підписаний будь-яким із користувачів системи. У деяких версіях DPoS делегат повинен показати зобов'язання, вкладаючи свої кошти в тимчасово заблокований захищений обліковий запис (які будуть конфісковані у випадку зловмисної поведінки); ця версія DPoS часто називається deposit-based Proof of Stake.

Delegated Proof of Stake не використовує умови звичайного Proof of Stake. Вибір виконується одним з наступних методів:

– делегати можуть бути обрані на основі їх частки в системі;

– делегати можуть отримувати голоси від усіх користувачів системи з правом голосу залежно від частки виборця;

– голоси делегатів за дійсні блоки можуть мати владу, пропорційну розміру їхнього депозиту.

В цілому, DPoS менш стандартизований, ніж базовий PoS [4].

1.2.4 Протокол консенсусу Proof of Activity

Протокол Proof of Activity є розвиненням протоколу Bitcoin. В PoA вузлам (англ. «nodes») потрібно проводити більш складні перевірки порівняно з тим, як це відбувається у Bitcoin, проте, існує думка, що ця додаткова робота дає певні переваги.

Першочерговим етапом, який включає PoA, є етап, який називається follow-the-satoshi, за допомогою якого деяке псевдовипадкове значення перетворюється в сатоші (найменша одиниця криптовалюти), який є рівномірно розподіленим серед всіх сатоші, які були отримані до даного часу. Це робиться шляхом вибору псевдовипадкового індексу в інтервалі від нуля до загальної кількості сатоші, що існують в системі аж до останнього блоку. При цьому виявляється блок, в якому даний сатоші був отриманий, і прослідковується ланцюг транзакцій, в процесі яких цей сатоші передавався між учасниками системи аж до його поточного власника (адреси, яка зараз контролює сатоші). Таким чином, можна вважати, що цей процес є псевдовипадковим рівномірним вибором серед учасників системи.

Далі буде описано процес генерації нового блоку в мережі PoA:

– Кожен майнер в системі використовує свою обчислювальну потужність, щоб спробувати згенерувати заголовок порожнього блоку – дані заголовка, які складаються з гешу попереднього блоку, адреси майнера в системі, висоти відносно блоку genesis та nonce (в залежності від значення якого змінюється значення гешу нового блока, що і визначає, чи буде даний блок дійсним). Даний заголовок не містить посилань на будь-які транзакції.

– Коли майнерові вдається створити заголовок порожнього блоку, це означає, що геш заголовку блоку є меншим, ніж поточна задана складність. В такому разі даний блок передається до мережі у якості

можливого кандидата.

– Для всіх вузлів мережі геш заголовку цього блоку є даними, які однозначно визначають N псевдовипадкових стейкхолдерів. Спочатку отримуються N псевдовипадкових значень шляхом об'єднання гешу даного блоку з гешом попереднього блоку і з N фіксованими суфіксами. Гешування кожної такої комбінації, а потім виклик follow-the-satoshi з кожним з гешів дає N стейкхолдерів.

– Кожен стейкхолдер, який наразі знаходиться в мережі, перевіряє, чи є заголовок блоку, що був переданий до мережі майнером, дійсним. Для цього перевіряється, що блок містить геш саме попереднього блоку та його геш є меншим за поточний встановлений поріг (складність). Після перевірки блоку користувач перевіряє, чи є він одним зі стейкхолдерів, обраних для цього блоку. Перші $N - 1$ стейкхолдери, яким адресовано блок, перевіряють його, підписують геш заголовка цього блоку своїми таємними ключами, які контролюють їх сатоші, і передають свої підписи до мережі. N-й стейкхолдер створює блок даних, який розширює заголовок порожнього блоку, включаючи до нього довільну кількість транзакцій, підписи $N - 1$ інших стейкхолдерів та свій власний підпис гешу всього цього блоку.

– N стейкхолдер передає блок до мережі, і коли інші вузли бачать, що даний блок є дійсним відповідно до описаного вище, вони вважають це законним доповненням блокчейну. Вузли намагаються розширювати найдовшу з точки зору складності гілку [5].

1.2.5 Протокол консенсусу Proof of Burn

Ідея даного протоколу полягає в тому, що для отримання можливості генерувати блоки, майнери повинні продемонструвати, що вони «спалили»

деякі монети (які можуть бути як валютою даного протоколу, так і іншою криптовалютою), тобто відправили їх на деяку адресу, з якої їх неможливо витратити чи повернути. Це є затратно з точки зору майнера, так само, як Proof of Work, але при такому підході не споживається жодних ресурсів, крім витрачених монет [6].

1.2.6 Протокол консенсусу Proof of Importance

Proof of Importance – алгоритм консенсусу, що застосовується у криптовалютній системі NEM. Суть даного протоколу полягає у присвоєнні кожному учаснику рейтингу, який формується на основі сукупного вкладу до економіки даної криптовалюти та, відповідно, важливості користувача системи. Користувачі з вищим рейтингом мають більшу імовірність приєднати блок транзакцій до ланцюга. Оскільки всі транзакції в системі є загальнодоступними для користувачів, граф транзакцій може бути використаний для обчислення рейтингу (важливості) користувача, що є головною рисою даної криптосистеми.

Створення нового блоку

Кожен блок складається з наступних елементів:

- версія блоку,
- мітка часу,
- відкритий ключ учасника, що створив даний блок (harvester),
- підпис даних блоку,
- геш даних попереднього блоку,
- геш для створення блоку (generation hash),
- висота блоку,
- перелік транзакцій.

Складність нового блоку обраховується на основі складностей та

міток часу попередніх 60 блоків. У випадку, коли блоків доступно менше, ніж 60, використовуються лише ті, що є. Якщо блок лише один, використовується складність за замовчуванням – 10^{14}. У іншому випадку, складність обраховується на основі попередніх n блоків наступним чином:

$$difficulty = d\frac{60}{t},$$

де

– d – середня складність блоків в ланцюгу,

$$d = \frac{1}{n}\sum_{i=1}^{n} difficulty\_of\_block(i),$$

– t – середній час створення блоку.

$$d = \frac{1}{n}\sum_{i=1}^{n} time\_to\_create\_block(i),$$

Якщо нова складність більш ніж на 5% більше або менше складності останнього блоку в ланцюгу транзакцій, то зміна складності обмежена верхньою межею, рівною 5%. Крім того, складність кожного блоку зберігається в певному проміжку. Нова складність фіксується в межах, якщо її значення становить більше 10^{15} або менше 10^{13}.

Процес створення нового блоку в системі NEM називається харвестинг (harvesting – збирання врожаю). Учасник, який додає новий блок, отримує винагороду з транзакцій, які перераховані у блоці. Будь-який учасник, що має на балансі принаймні 10000 валютних одиниць XEM, може бути допущеним до харвестингу. Для перевірки можливості створення нового блоку учасником, обраховуються наступні величини:

– $h = H(generation\_hash$ попереднього блоку, відкритий ключ користувача), де H – 256-бітна геш-функція SHA3,

– t – час в секундах з моменту приєднання останнього блоку,

– $b = 8999999999 \cdot$(рейтинг облікового запису),

– d – складність нового блоку.

На основі їх отримують дві величини - hit (англ. «hit» – потрапляння) та $target$ (англ. «target» – ціль, мета):

$$hit = 2^{54}\left|ln\left(\frac{h}{2^{256}}\right)\right|,$$

$$target = 2^{64}\frac{b}{d}t,$$

Користувачеві, який першим отримав значення hit та $target$ такі, що

$$hit < target$$

надається можливість створити новий блок.

Оскільки величина $target$ пропорційна пройденому часу, можливість згенерувати новий блок буде навіть у тому випадку, коли всі користувачі системи отримали дуже велике значення hit. Також варто зазначити, що величина hit має експоненціальний розподіл, що означає незмінність імовірності створення нового блоку при розподіленні важливості між багатьма учасниками одного кластера.

Обчислення рейтингу користувача

Розглянемо процес, необхідний для підрахунку рейтингу користувача в термінах даної криптовалютної системи. Припустимо, що обчислення Proof of Importance виконуються на висоті h. Користувач, який має значення активного балансу не менше за 10000 XEM (атомарних валютних одиниць) на висоті h, має право приймати участь у даному обчисленні. Для таких користувачів NEM збирає всі їх транзакції передачі, які задовольняють наступним умовам:

– передавалася сума не менше 1000 XEM,

– транзакція проводилась в межах останніх 43200 блоків (приблизно 30 днів),

– одержувач суми також має право брати участь в обчисленні.

Для кожної такої транзакції T_k, в ході якої передається $amount$ ХЕМ від користувача A_i до користувача A_j на висоті h_{ijk} вага обраховується наступним чином:

$$w_{ijk} = amount \cdot exp\Big(ln(0{,}9)\Big[\frac{h - h_{ijk}}{1440}\Big]\Big),$$

де $[x]$ позначає функцію отримання цілої частини. Ці значення збираються в суму

$$\tilde{w}_{ij} = \sum_k w_{ijk}.$$

Встановивши

$$\tilde{o}_{ij} = \begin{cases} \tilde{w}_{ji} - \tilde{w}_{ij} & \text{, якщо } \tilde{w}_{ji} - \tilde{w}_{ij} > 0 \\ 0 & \text{, інакше} \end{cases}$$

отримуємо матрицю O, елементами якої є

$$o_{ij} = \begin{cases} \frac{\tilde{o}_{ij}}{\sum_i \tilde{o}_{ij}} & \text{, якщо } \sum_i \tilde{o}_{ij} > 0 \\ 0 & \text{, інакше} \end{cases}$$

Таким чином, елемент o_{ij} описує зважений чистий потік ХЕМ від користувача A_i до A_j протягом (приблизно) 30 днів. Це означає, що лише чисті переведення сприяють рейтингу користувача.

Для визначення значимості учасників використовується алгоритм $NCDawareRank$, який ітеративно обчислюється за наступною формулою:

$$\tilde{\pi} = O\eta\pi + M\mu\pi + E(1 - \eta - \mu)\pi,$$

де:

– O – матриця вихідних зв'язків, описана вище,

– M – міжрівнева матриця близькості,

– E – матриця телепортації,

– π – $NCDawareRank$,

– η – частина рейтингу важливості, яка присвоюється на основі вихідних зв'язків,

– μ – частина рейтингу важливості, яка присвоюється найближчим обліковим записам.

Розрахунок рейтингу облікових записів виконується за наступною формулою:

$$\phi = (normalize(max(0, \nu + \sigma\omega_0)) + \tilde{\pi}\omega_1)\chi,$$

де:

– $normalize(\nu) = \frac{\nu}{||\nu||}$,

– ν – кількість ХЕМ на балансі учасника,

– σ – зважений чистик потік ХЕМ,

– $\tilde{\pi}$ – оцінка $NCDawareRank$,

– χ – вектор вагів, який враховує структурну топологію графа,

– ω_0, ω_1 – константи [7].

1.3 Протоколи розділення секрету

Розподіл секрету – термін в криптографії, під яким розуміють будь-який із способів розподілу деякого секрету між групою учасників, кожному з яких дістається своя частина. Секрет можна відтворити тільки при наявності коаліції учасників первісної групи, при чому кількість учасників коаліції має бути не меншою деякого заданого числа [8].

У найпростішому випадку секретне повідомлення ділиться між двома особами. Поділ секрету проводить окрема особа – розподілювач. Для цього виконуються наступні кроки:

– розподілювач генерує вектор випадкових бітів R, довжина якого співпадає з довжиною секретного повідомлення,

– розподілювач обчислює $S = R \oplus M$,

– першій особі передається значення R, а другій – S.

Для відновлення повідомлення достатньо обчислити наступне:

$$M = R \oplus S.$$

Даний метод при правильному виконанні є абсолютно безпечним. Кожна частина розділеного секрету не несе ніякої змістовної інформації. Дійсно, розподілювач шифрує повідомлення одноразовим блокнотом і роздає одній особі шифр, іншій блокнот, тобто, ніякі обчислювальні потужності не допоможуть відновити повідомлення лише по одній його частині. Головним недоліком такого підходу до розподілу є те, що втрата однієї з частин секрету та відсутність повідомлення у розподілювача призведе до його втрати, адже особам, між якими його розподілено, нічого, крім довжини засекреченого повідомлення, про нього не відомо.

Для уникнення такого недоліку і для розширення можливостей використання протоколів, А. Шаміром та Дж. Блеклі незалежно було запропоновано сумісне використання секрету учасниками розподілу, яке отримало назву (m, n)-порогова схема. Суть даного підходу полягає в тому, що секрет розподіляється між n особами таким чином, що будь-які $m \leq n$ з них (дозволена коаліція) можуть відновити повідомлення, пред'являючи свої частини, а будь-які $(m-1)$ – ні.

Даний напрямок інтенсивно вивчався Г. Сіммонсом, в результаті чого були запропоновані модифікації порогової схеми, серед яких наступні:

– схеми, що забезпечують коректність роботи при наявності шахраїв серед учасників (в тому числі і у випадку, коли шахраєм є розподілювач),

– схеми, які не передбачають розподілювача,

– схеми, в яких не розкриваються розділені частини секрету (що дозволяє їх повторно використовувати),

– схеми з перевіркою секрету, які дозволяють кожному з учасників переконатися в правильності його частини,

– схеми з запобіжними заходами, які дозволяють одній частині учасників завадити дозволеній коаліції розкрити секрет,

– схеми з викресленням зі списку, для яких існують способи копіювання, які забезпечують активацію нового кола учасників без перерозподілу секрету (що необхідно у випадках, коли деякі з учасників скомпрометовані).

Розглянемо алгоритми, які реалізують описані ідеї.

1.3.1 Схема Блеклі

Дж. Блеклі винайшов схему, яка використовує поняття точок в багатовимірному просторі.

Нехай потрібно розподілити секретне повідомлення $m \in Z_p$ між n учасниками таким чином, щоб будь-які t з них могли відновити повідомлення, об'єднавши свої частини секрету, а будь-які $(t-1)$ – не мали можливості нічого дізнатися про m. Повідомлення представляється у вигляді точки в t-мірному просторі, а кожна частина секрету – рівняння (t - 1)-мірної гіперплощини, яка містить дану точку [9].

Розглянемо алгоритм на прикладі $(3, n)$-порогової схеми. Розподілювач обирає p – просте число, яке більше всіх можливих секретів. Нехай секрет - $x_0 \in Z_p$. Випадковим чином обираються точки $y_0, z_0 \in Z_p$. Таким чином формуємо секретну точку $Q = (x_0, y_0, z_0)$. Формування частини секрету для кожного учасника відбувається наступним чином:

– обираються випадкові коефіцієнти $a, b \in Z_p$,

– обчислюється значення $c = z_0 - a \cdot x_0 - b \cdot y_0$,

– отримуємо площину: $z = a \cdot x + b \cdot y + c$, яка і є частиною секрету конкретного учасника.

Для отримання повідомлення знаходиться точка перетину площин, рівняння яких мають 3 учасники. Зрозуміло, що при кількості учасників, меншій, ніж 3, можна отримати лише пряму, на якій знаходиться шукана точка, що робить її знаходження неможливим.

1.3.2 Схема Шаміра

Для створення порогової схеми А. Шамір скористався рівняннями поліномів у скінченному полі та інтерполяційною формулою Лагранжа.

Нехай p – просте число, яке більше всіх можливих секретів. Потрібно розподілити секретне повідомлення $m \in Z_p$ між n учасниками таким чином, щоб будь-які t з них могли відновити повідомлення, об'єднавши свої частини секрету, а будь-які $(t-1)$ – не мали можливості нічого дізнатися про m. Для цього розподілювач фіксує та публікує серед учасників значення p та випадково обирає набір коефіцієнтів $s_1, \ldots, s_{t-1} \in Z_p$, на основі яких будується поліном:

$$s(x) = m + s_1 \cdot x + \ldots + s_{t-1} \cdot x^{t-1}.$$

Коефіцієнти зберігаються в таємниці та знищуються після розподілу секрету. Розподіл секрету відбувається шляхом розсилання кожному i-му $(i = 1, ..., n)$ учаснику пари $(i, s(i))$, де $s(i)$ - значення поліному в точці i.

Як зазначалось раніше, для відновлення секрету потрібна участь принаймні t осіб. Разом вони мають набір значень полінома $((x_1, s(x_1)), ..., (x_t, s(x_t))$. Очевидно, що секретом є $m = s(0)$. Згідно з формулою Лагранжа, отримуємо значення m:

$$m = \sum_{i=1}^{t} s(x_i) \frac{\prod_{j \neq i}^{j \in [1,\ldots,t]} -x_j}{\prod_{j \neq i}^{j \in [1,\ldots,t]} x_i - x_j}.$$

Альтернативний спосіб знаходження секрету – розв'язування системи лінійних рівнянь наступного вигляду:

$$\begin{pmatrix} 1 & x_1 & \ldots & x_1^{t-1} \\ 1 & x_2 & \ldots & x_2^{t-1} \\ \vdots & & \vdots & \vdots \\ 1 & x_t & \ldots & x_t^{t-1} \end{pmatrix} \begin{pmatrix} m \\ s_1 \\ \vdots \\ s_{t-1} \end{pmatrix} = \begin{pmatrix} s(x_1) \\ s(x_2) \\ \vdots \\ s(x_{t-1}) \end{pmatrix}$$

Як відомо, якщо всі значення $x_1, \ldots, x_t$ різні (а вони є різними, оскільки їх встановлювали як індекси користувачів), то визначник такої матриці є ненульовим i, відповідно, система має єдиний розв'язок. Розв'язавши її, в числі невідомих отримуємо значення повідомлення m.

Схема Шаміра має наступні основні властивості:

– абсолютна секретність (теоретико-інформаційна стійкість): наявність будь-яких $k < t$ частин секрету не дає жодної інформації про секрет,

– ідеальність: кількість бітів, що містить кожна частина секрету, рівна кількості бітів самого секрету,

– розширюваність: кількість власників частин секрету може бути збільшена до $|Z_p| - 1$, при цьому кількість частин секрету, необхідних для його відновлення, залишається незмінною,

– гнучкість: є можливість надавати різним користувачам різні «ваги» – в залежності від важливості учасника надаються різні кількості частин секрету.

Недоліками такого підходу є одноразовість, можливість шахрайства з боку учасників та розподілювача [10].

1.3.3 Схема Асмута - Блума

Схема базується на китайській теоремі про лишки. Для (m, n)-порогової схеми обирається велике просте число p, яке більше за повідомлення M. Потім обираються числа, менші за p: $d_1, d_2, \ldots, d_n$, для яких виконуються наступні умови:

– значення d_i впорядковані за зростанням,

– для будь-якого $i \neq j : gcd(d_i,\ d_j) = 1$,

– $d_1 \cdot d_2 \cdot ... \cdot d_m < p \cdot d_{n-m+2} \cdot ... \cdot d_n$.

Для розділення секрету на частини обирається випадкове число r та обчислюється $M^{'} = M + rp$. В такому випадку частинами секрету є набір значень $k_i = M^{'}(mod\, d_i), i = 1, ..., n$.

Поєднавши будь-які m частин, можна відновити значення повідомлення M, використовуючи китайську теорему про лишки, але неможливо відновити його значення, маючи лише $(m - 1)$ частин [11].

1.3.4 Схеми розподілу з перевіркою секрету

Припускаючи, що розподілювач секрету може виявитися шахраєм або частина секрету деякого учасника пошкодилась в процесі її передачі по каналу зв'язку, виникає необхідність механізму перевірки учасником своєї долі на правильність. В схемах, описаних вище, єдиний спосіб переконатися у правильності – спробувати відновити секрет m. Враховуючи одноразовість такого розподілу та потенціальну небажаність розкриття учасниками їх долей для інших, були розроблені схеми, які дозволяють кожному з учасників особисто переконатися у правильності його частини. Серед них найбільш розповсюдженими є схеми Педерсона

та Фельдмана.

Схема Фельдмана-Шаміра

Фельдман запропонував верифікацію схеми Шаміра, стійкість якої базується на основі складності обчислення дискретного логарифма у скінченних полях.

Нехай $A_1, ..., A_n$ - учасники схеми, D - розподілювач. p, q – великі прості числа, при чому $q|(p-1)$, g - деякий елемент мультиплікативної групи Z_p^* порядку q. В даному випадку у якості поля буде використовуватися Z_q. Після побудови розподілювачем полінома над полем, він обчислює значення $r_i = g^{s_i}(mod\, p), i = 1, ..., (t-1)$, де $s_0 = m$. Після цього значення r_i розміщуються у відкритому доступі. Складність обчислення значень s_i за відомими значеннями r_i базується на складності задачі дискретного логарифмування у скінченному полі.

Після отримання учасником A_i частини секрету $(x_i,\ s(x_i))$ він може переконатися у його правильності, перевіривши наступне порівняння:

$$g^{s(x_i)} \equiv r_0(r_1)^{x_i}...(r_{t-1})^{x_i^{t-1}}(mod\, p).$$

Також після відновлення секрету m будь-якою групою з t учасників можна перевірити його дійсність за допомогою наступного порівняння:

$$g^m \equiv r_0(mod\, p).$$

[12]

Схема Педерсона-Шаміра

На відміну від схеми Фельдмана, наведена нижче схема має теоретико-інформаційну стійкість (абсолютну стійкість). Нехай числа p, q, g, m визначені аналогічно попередній схемі, $d \in Z_q$ - деякий секретний параметр, $h = g^d(mod\, p)$ – відкритий параметр.

Для розділення секрету $m \in Z_q$ обираються два поліноми:

$$P(x) = m + a_1 \cdot x + ... + a_{t-1} \cdot x^{t-1},$$

$$Q(x) = b_0 + b_1 \cdot x + ... + b_{t-1} \cdot x^{t-1},$$

де a_i, b_j – випадкові числа з Z_q, $i = 1, ..., t-1, j = 0, ...,t-1$ та обчислює значення $y_k = P(x_k)$, $z_k = Q(x_k)$, $k = 1, ...,n$. Частинами секрету, що розподіляються між учасниками є трійки (x_i, y_i, z_i), $i = 1, ...,n$. Для перевірки коректності частин секрету розподілювач публікує значення:

$$r_i = g^{a_i} h^{b_i} (mod\, p), i = 0, ..., t-1.$$

Важливим є те, що, на відміну від схеми Фельдмана, значення $r_0 = g^{m+db_0} (mod\ p)$ залежить від випадкового числа b_0, що робить неможливим отримання будь-якої інформації про секрет навіть за умови обчислення значень d та $m + db_0 (mod\ q)$ (вирішення задачі дискретного логарифмування).

Тепер кожен учасник може перевіряти дійсність своєї частини секрету, виконавши перевірку порівняння:

$$g^{y_i} h^{z_i} \equiv r_0 (r_1)^{x_i} ... (r_{t-1})^{x_i^{t-1}} (mod\, p),$$

оскільки

$$g^{y_i} h^{z_i} = g^{P(x_i)} h^{Q(x_i)} = g^m g^{a_1 x_i} ... g^{a_{t-1} x_i^{t-1}} h^{b_0} h^{b_1 x_i} ... h^{b_{t-1} x_i^{t-1}} =$$
$$= (g^m h^{b_0})(g^{a_1} h^{b_1})^{x_i} ... (g^{a_{t-1}} h^{b_{t-1}})^{x_i^{t-1}} (mod\, p) \equiv r_0 (r_1)^{x_i} ... (r_{t-1})^{x_i^{t-1}} (mod\, p).$$

Також після відновлення секрету m будь-якою групою з t учасників можна перевірити його дійсність за допомогою наступного порівняння:

$$g^m h^{b_0} \equiv r_0 (mod\, p),$$

[13]

Висновки до розділу 1

У даному розділі було розглянуто ключові елементи побудови криптовалютних систем, серед яких наведено та досліджено технологію «блокчейн», описано деякі найбільш популярні та використовувані протоколи досягнення узгодження, розглянуто їх переваги та недоліки. Також було описано існуючі порогові протоколи розподілу секрету з метою їх подальшого застосування до побудови модифікованих протоколів узгодження. Надалі буде запропоновано внесення додаткових умов в існуючий протокол з метою усунення певних недоліків.

2 ПОБУДОВА МОДИФІКАЦІЙ ПРОТОКОЛІВ КОНСЕНСУСУ

В розглянутих раніше протоколах консенсусу є один суттєвий недолік – ресурс, за яким визначається рейтинг учасника при формуванні угоди прямо чи непрямо пов'язаний із деяким реальним цінним ресурсом. В протоколі Proof of Work це час, обчислювальні ресурси та значні витрати електроенергії; протокол Proof of Stake влаштований таким чином, щоб надавати перевагу генерування нового блоку користувачам, які мають більший відсоток валюти в системі. Також існують протоколи, в яких цінним ресурсом виступають пам'ять, активність учасника в системі чи його репутація. Інша проблема – присутність у протоколах механізмів, які обмежують можливість підвищення власного рейтингу вище деякого порогу (для кожного окремого учасника такий поріг визначається окремо). Таким чином, постає проблема того, що багатші стають багатшими, не даючи можливість іншим зацікавленим учасникам проводити транзакції щодо додавання нових блоків до системи та отримувати з цього вигоду. Перша з наведених нижче модифікацій протоколу консенсусу дозволяє періодично певним чином проводити перерозподіл учасників, які мають право згенерувати новий блок, а друга – періодично отримувати перевагу над масштабним гравцями в системі шляхом наявності іншого цінного ресурсу та отримання знання, що надає перевагу.

2.1 Лотерея

Протокол базується на описаній вище схемі досягнення консенсусу Proof of Importance криптовалюти NEM. Як зазначалось, для отримання права генерувати нові блоки, першочергово обліковий запис учасника має задовольняти умові наявності на балансі принаймні 10000 валютних одиниць, проте це може дозволити собі не кожен учасник. Дане рішення націлене на часткове усунення цієї проблеми шляхом періодичного обрання кола учасників та надання їм інформації, що робить їх облікові записи спроможними приєднувати до ланцюга нові блоки. Таким чином, протокол націлений на надання шансу збільшити імовірність успіху певного кола користувачів, але не дає однозначної гарантії перемоги в даному періоді.

Назвемо періодом T_j проміжок часу між приєднаннями до ланцюга висоти j блоку з номером $(j+1)$. Наприкінці кожного періоду учасник, який виконав приєднання, має можливість розподілити по системі інформацію, що надасть можливість згенерувати блок, незважаючи на поріг входження до даного кола користувачів. Цей процес відбувається шляхом генерації випадкового секретного значення, що представляє собою деяке $m \in Z_p$, де p – велике просте число.

Наступним етапом є вибір адрес учасників. На даному етапі протокол можна розділити на два подальші напрямки розвитку в залежності від того, яким чином адреси будуть обиратися. Розглянемо ці напрямки.

Криптовалютна система NEM передбачає використання відкритих ключів користувачів, які представлені цілими числами довжини 256 бітів. На основі цих ключів генеруються дані, які використовуються у якості адрес користувачів. У найпростішому випадку, вибір адрес полягає у генерації з перевіркою існування в системі випадкових 256 бітних значень $x_i, i = 1, ...,k$, де k становить деякий відсоток від загальної кількості

учасників системи. Кількість частин, необхідних для відновлення секрету в даному випадку не є важливою, оскільки саме його значення ніяк не фігурує у процесі набуття права генерації, тому покладемо це значення рівним k.

Секрет m розділяється на k частин за допомогою схеми Педерсона-Шаміра, в результаті чого отримуємо набір трійок значень $(x_i, y_i, z_i), i = 1, ...,k$, які розсилаються за попередньо обраними адресами. Разом з цим розподілювач також приєднує до нового блоку набір значень $r_j, j = 0, ...,k-1$ та значення g і h, що в подальшому дасть можливість учасникам, які отримали частини секрету, підтвердити публічно їх достовірність.

Оскільки адреси обиралися без перевірки рейтингу користувачів за даними адресами, можливі наступні випадки:

– частину секрету отримає учасник без права приєднання блоку,

– частину секрету отримає учасник з правом приєднання блоку.

Відповідно, в обох випадках учасник повинен отримати певну перевагу у генерації таких значень hit та target, які задовольнятимуть нерівності:

$$hit < target,$$

що і означає надання права приєднання блоку. Розглянемо кожен з цих випадків окремо.

Нехай учасник A_i, який отримав частину секрету (x_i, y_i, z_i), не має права згенерувати новий блок, тобто, значення його балансу менше 10000 валютних одиниць XEM. В такому випадку, його рейтинг, згідно зі специфікацію NEM, вважається нульовим, що робить неможливим отримання пари значень $hit < target$ згідно з побудовою значення $target$. Оскільки значення $hit = 2^{54}\left|ln\left(\frac{h}{2^{256}}\right)\right|$ за своєю побудовою не залежить від параметрів системи чи користувача, доцільно змінювати значення $target$ шляхом заміни на один період рейтингу даного користувача деяким ненульовим значенням.

Після отримання частини секрету учасник A_i повинен повідомити інших учасників системи про це та надати доказ дійсності цієї частини шляхом публікації значень трійки (x_i, y_i, z_i). Інші учасники можуть переконатися у коректності цих значень шляхом перевірки рівності:

$$g^{y_i} h^{z_i} \equiv r_0 (r_1)^{x_i} ... (r_{t-1})^{x_i^{t-1}} (mod\, p).$$

У разі затвердження для даного користувача його рейтинг переобчислюється шляхом заміни його фактичного балансу на мінімальне значення, необхідне для допуску до харвестингу, в результаті чого він отримує ненульове значення $target$, що дозволяє йому прийняти участь у процесі вибору наступного харвестера, тобто, в нього з'являється шанс згенерувати новий блок.

Розглянемо випадок, коли частину секрету отримав учасник з правом приєднання блоку. В такому випадку, він має ненульове значення рейтингу, що означає, що він, незалежно від наявності долі секрету, може отримати пару значень (hit, $target$), які задовольняють наведеній вище нерівності. Тому доцільно даному учаснику надавати можливість збільшити імовірність успішної генерації. Очевидно, що для цього достатньо певним чином пропорційно збільшити значення величини $target$.

Аналогічно попередньому випадку, після отримання частини секрету учасник A_i повинен повідомити інших учасників системи про це та надати доказ дійсності цієї частини шляхом публікації значень трійки (x_i, y_i, z_i). Інші учасники можуть переконатися у коректності цих значень шляхом перевірки рівності:

$$g^{y_i} h^{z_i} \equiv r_0 (r_1)^{x_i} ... (r_{t-1})^{x_i^{t-1}} (mod\, p).$$

У разі затвердження даний користувач отримує право переобчислити значення $target$ як:

$$target = 2^{64}\frac{b \cdot c}{d}t,$$

де значення c залежить від долі секрету та визначається наступним чином:

$$c = ln(y_i + z_i).$$

Головним недоліком такого підходу є те, що поточний харвестер має можливість певним чином замінити випадковий вибір адрес на вибір конкретних адрес, у перемозі яких він зацікавлений (в тому випадку, якщо програмна реалізація має помилки безпеки).

Другий напрямок передбачає використання підходу, схожого до протоколу Proof of Activity. Головною особливістю даного підходу є принципова неможливість будь-яким чином попередньо визначити адреси користувачів, які отримають секрет.

Нехай харвестер H_i вирішив розподілити секрет в системі. Для цього генерується деяке псевдовипадкове значення $m \in Z_p$. У результаті його розділення на k частин, згідно зі схемою Передсона-Шаміра, також обчислюється набір значень $r_0, \ldots, r_{k-1}$, що приєднується до останнього блока в ланцюгу. Для подальшого опису необхідно внести деякі зміни до структури валютних одиниць системи XEM. Згідно зі специфікацією, загальна кількість XEM в системі становить 8999999999. Зважаючи на це, існує можливість проіндексувати їх.

Маючи у публічному доступі значення гешу останнього блоку $hash_j$, гешу попереднього (передостаннього) блоку $hash_{j-1}$ та значень $r_i, i = 0, ...,(k-1)$, можемо сформувати k індексів:

$$A_i = H(hash_j \,||\, hash_{j-1} \,||\, r_{i-1})(mod\, 8999999999),$$

де H – 256-бітна геш-функція $SHA3$. На основі отриманих індексів обираються облікові записи учасників, які мають XEM з номером, що відповідає одному з індексів. Саме ці користувачі будуть обрані для розподілення секрету. Такий підхід дозволяє повністю виключити

можливість компрометації вибору адрес, оскільки навіть на етапі отримання значень $r_0, \ldots, r_{k-1}$ обчислювально складно підібрати значення коефіцієнтів поліномів таким чином, щоб отримати бажані значення (вирішення задачі дискретного логарифмування та дуже малий час), а поведінку обраної геш-функції передбачити практично неможливо.

Подальший етап поведінки учасників даного протоколу аналогічним тому, що був описаний раніше.

2.2 Використання IP адрес у якості цінного ресурсу

Дана модифікація протоколу націлена на використання у якості цінного ресурсу IP адрес наявних у кожного окремого користувача фізичних пристроїв. Такий підхід робить протокол схожим на Proof of Stake, але при цьому він є значно менш ресурсоємним з точки зору витрат електроенергії. Суть даного протоколу полягає у наданні додаткової переваги перед іншими учасниками з правом генерації блоку з великою кількістю пристроїв, підключених до одного облікового запису. Такий підхід стимулює зростання активності окремих груп учасників в економіці системи.

Аналогічно попередньо описаному протоколу, введемо поняття періоду T_i як проміжок часу між приєднаннями до ланцюга висоти i блоку $(i + 1)$. Наприкінці кожного періоду учасник, який згенерував новий блок, має можливість розподілити по системі інформацію, що надає перевагу у генеруванні блоку. Цей процес відбувається шляхом обрання випадкового секретного значення, що представляє собою деяке $m \in Z_p$, де p – велике просте число.

Нехай в системі зареєстровано n учасників. В даному випадку будемо розглядати модифікацію системи, яка дозволяє приєднувати до

облікового запису більше одного фізичного пристрою. Покладемо, що обліковий запис учасника i має x_i підключених пристроїв, у кожного з яких, відповідно, своя унікальна IP адреса. Отже, в системі є $n\sum_i x_i$ унікальних адрес. Величина значення p обирається таким чином, щоб $n\sum_i x_i < |Z_p| - 1$.

Розподілення секрету відбувається між всіма наявними активними фізичними пристроями системи таким чином, щоб кожен міг підтвердити коректність своєї отриманої частини. Для цього учасник, що приєднав до ланцюга останній блок, згідно з схемою Передсона-Шаміра, генерує набір значень (x_i, y_i, z_i), де значення x_i є унікальним та відповідає деякій адресі в системі та публікує разом з останнім приєднаним блоком набір значень, які дадуть можливість перевірити коректність секрету в подальшому, саме значення $g \in Z_P^*$, $ord(g) = q$, $h = g^d (mod\, p)$ та $r_0 = g^{m+db_0} (mod\, p)$, де $d \in Z_q$ – деякий секретний параметр.

Як було описано раніше, в криптовалютній системі NEM учасник, який має право генерації блоку, оперує наступними двома величинами:

$$hit = 2^{54} \left| ln(\frac{h}{2^{256}}) \right|,$$

$$target = 2^{64} \frac{b}{d} t.$$

Відповідно, особа, якій вперше вдалось згенерувати пару значень $(hit, target)$ таку, що $hit < target$ та поширити ці значення в системі з метою підтвердження, надається право на генерацію. Варто зазначити, що вплинути на значення hit та $target$ в конкретний момент часу та на конкретній висоті ланцюга блоків транзакцій неможливо, оскільки значення hit залежить лише від попереднього блоку та відкритого ключа користувача, а зміна значення $target$ вимагає очікування або навіть зміни висоти ланцюга. Згідно зі специфікацією NEM [7], можливі випадки, коли в певні проміжки часу жоден з учасників системи з правом генерації блоку не може цього зробити, оскільки в даний конкретний момент

неможливо отримати необхідні для цього значення. Даний конфлікт вирішується, оскільки час t між приєднаннями двох сусідніх блоків, значення якого прямопропорційно впливає на значення $target$, постійно зростає. Але протягом даного часу є можливість змінити імовірність успішної генерації пари $(hit, target)$, яка буде задовольняти умові отримання права генерації. Один з можливих варіантів реалізації – пропорційна зміна значення величини hit або $target$ таким чином, щоб збільшити імовірність потрапляння hit в інтервал $[0, target)$. У зв'язку з цим необхідно розглянути наступні питання:

– За яким критерієм надавати право на зміну значень пари $(hit, target)$?

– Яким чином змінювати дані значення?

– Як дана модифікація вплине на розподіл права генерації між учасниками системи?

Розглянемо ці питання.

При описаному підході до побудови системи очевидним критерієм для надання права на зміну значень пари $(hit, target)$ є публічне підтвердження володіння секретом, згенерованим останнім харвестером (учасником системи, що згенерував блок). В цьому полягає суть IP адрес як цінного ресурсу в даній схемі – чим більше пристроїв містить конкретний обліковий запис, тим більше його шанс періодично отримувати можливість збільшувати імовірність згенерувати наступний блок. Згідно зі схемою Педерсона-Шаміра, є певне порогове значення k кількості секретів, володіння якими необхідне для відновлення початкового секретного значення. Значення k обирається окремо при кожному розповсюдженні секрету в системі. Відповідно, якщо учасник A_i має $x_i \geq k$ пристроїв, під'єднаних до облікового запису, він може відновити секрет та отримати перевагу. Для підтвердження правильності відновленого секрету він публікує значення секрету m та значення b_0, на основі чого будь-який учасник може перевірити його коректність за формулою:

$$g^m h^{b_0} \equiv r_0 (mod\, p)$$

Розглянемо способи зміни значень пари $(hit, target)$. Зважаючи на нерівність, що застосовується для оцінки можливості приєднання блоку до ланцюга:

$$hit < target,$$

доцільно змінювати значення лише одного з параметрів, оскільки цього буде достатньо для отримання переваги користувачем, що зміг відновити секрет m. Модифікацію значення можна проводити на основі одного з наступних факторів:

– значення секрету m,

– деяке порогове значення, що гарантує перевагу,

– рейтинг учасника.

Числове значення m може використовуватися у якості складової частини обчислення значення параметра hit. За припущення, що кожен згенерований секрет матиме значення $m > 3$, будемо допускати в даному протоколі обчислення hit наступним чином:

$$hit^{'} = \frac{2^{54}}{ln(m)} \left| ln(\frac{h}{2^{256}}) \right|.$$

Таке перетворення зменшує значення параметра hit. Дійсно, оскільки у якості секрету було обрано значення $m > 3$, значення натурального логарифму завжди буде додатнім числом зі значенням > 1. З припущення, що число p буде довжини 1024 біти, а $m \in Z_p$, таке перетворення може значним чином – навіть у 2^{10} разів – зменшити значення hit. Це надає значну перевагу користувачу, який спромігся відновити значення секрету.

Недоліком такого підходу є те, що навіть при отриманні значення секрету існує ненульова імовірність того, що знайдеться учасник з більшим значенням рейтингу, який навіть без відновлення секрету буде

спроможний отримати пару значень, яка задовольняє нерівності ($hit < target$).

Дана модифікація протоколу дозволяє зменшити час простою системи у випадку, коли жоден з учасників не зміг отримати необхідну пару значень, що позитивно впливає на швидкість проходження транзакцій в системі, та, як наслідок, на швидкість розвитку економіки криптовалютної системи, а також стимулює учасників до залучення нових обчислювальних ресурсів.

Висновки до розділу 2

В даному розділі було запропоновано два підходи щодо можливих модифікацій існуючих протоколів консенсусу в розподілених криптовалютних системах. В основі обох підходів лежить концепція розподілення секрету між випадковою групою учасників, що надає їм додаткове знання, і, як наслідок, перевагу перед іншими обліковими записами.

Перший підхід описує можливість використання принципу лотереї в такій системі, що робить періодично можливим створення та приєднання нового дійсного блоку до ланцюга, та, як наслідок, отримання комісії за це учасниками, які, згідно з побудовою системи, не мають на це права.

Другий підхід орієнтований лише на учасників з правом генерації блоку та орієнтований на перерозподіл даного права протягом деяких періодів з метою уникнення зосередження його в межах певної групи.

Надалі буде проведено порівняльний аналіз існуючих рішень порівняно з запропонованими, а також наведено оцінки імовірностей згенерувати новий блок у кожному з протоколів.

3 АНАЛІЗ ПРОТОКОЛІВ КОНСЕНСУСУ

В даному розділі буде наведено теоретичні оцінки імовірностей генерації нового блоку в блокчейні для кожного з описаних раніше протоколів консенсусу в залежності від кількості та типу цінного ресурсу учасника. Це дозволить виявити залежність між типом та кількістю наявного в учасника цінного ресурсу та його практичною можливістю згенерувати новий дійсний блок в криптосистемі. На основі такого підходу можна показати, що запропонована модифікація дійсно збільшує шанси учасників, які отримують додаткову інформацію. Розглянемо дані оцінки.

3.1 Proof of Work

Нехай B – запропонований новий блок, який вважається дійсним в рамках даного протоколу, тобто,

$$hash(B) \leq M/D,$$

де $D \in [1,\ M]$ – складність даного блоку. Введемо значення $T(r)$ як кількість часу в секундах, необхідного для генерації блоку B майнером з апаратним забезпеченням, здатним виконувати r операцій в секунду.

Припустимо, що всі дійсні блоки в протоколі консенсусу задовольняють умові:

$$U \leq \theta \leq 1,$$

де $U \sim Un[0, 1]$ – рівномірно розподілена випадкова величина, отримана

шляхом гешування певних даних та нормалізації даної величини таким чином, щоб її значення знаходилось на проміжку $[0, 1]$. Proof of Work – частинний випадок даної умови зі значенням $\theta = \frac{1}{D}$.

Для успішної генерації блоку в термінах описаних вище позначень, користувач має знайти вхідні дані, значення яких відповідає значенню U, що задовольняє нерівності $U \leq \theta$. Нехай N – кількість варіантів таких даних, які необхідно перебрати перед тим, як буде отримане необхідне значення вхідних даних. Тоді можемо визначити $T(r)$ наступним чином:

$$T(r) = \frac{N}{r}.$$

Розглянемо розподіл даної величини:

$$Pr\{T(r) \leq t\} = Pr\{\frac{N}{r} \leq t\} = Pr\{N \leq tr\} = 1 - Pr\{N > tr\}.$$

У випадку рівномірного розподілу імовірність вибрати значення, яке задовольняє нерівності, становить $\theta = \frac{1}{D}$, відповідно, неправильне – $(1 - \frac{1}{D})$. Тоді $Pr\{N > tr\} = \left(1 - \frac{1}{D}\right)^{tr}$. В такому разі можемо продовжити обчислення:

$$Pr\{T(r) \leq t\} = 1 - Pr\{N > tr\} = 1 - \left(1 - \frac{1}{D}\right)^{tr} = 1 - exp\Big(ln\Big(1 - \frac{1}{D}\Big) \cdot tr\Big).$$

Зважаючи на величину значення D, можемо допускати, що $\frac{1}{D} << 1$, тому $ln\Big(1 - \frac{1}{D}\Big) \approx -\frac{1}{D}$. Тоді

$$Pr\{T(r) \leq t\} = 1 - exp\Big(-\frac{tr}{D}\Big).$$

Нехай в криптовалютній системі є n майнерів з частотами r_1,...,r_n. Тоді час знаходження дійсного блоку B становить $T_b = min(T_1,...,T_n)$, де T_i – час, необхідний для обчислення значення i-му майнерові. Тоді:

$$Pr\{T_b \leq t\} = 1 - Pr\{min(T_1,...T_n) \ > t\} = 1 - Pr\{T_1 > t, ..., T_n > t\} =$$

$$= 1 - \prod_{i=1}^{n} Pr\{T_i > t\} = 1 - \prod_{i=1}^{n} exp\Big(-\frac{tr_i}{D}\Big) = 1 - exp\Big(-\frac{t}{D}\sum_{i=1}^{n} r_i\Big)$$

Отже, бачимо, що час, необхідний для генерації нового дійсного блоку розподілений експоненціально з параметром $\lambda = -\frac{1}{D}\sum_{i=1}^{n} r_i$. Тоді за властивістю експоненціального розподілу, отримуємо імовірність згенерувати дійсний блок майнером i:

$$Pr\{T_b = T_i\} = \frac{r_i}{\sum_{j=1}^{n} r_j}.$$

Це означає, що майнінг у протоколі Proof of Work є справедливим, тобто, майнер з часткою потужності p має імовірність p згенерувати наступний дійсний блок.

3.2 Proof of Stake

Нехай B – запропонований новий блок, який вважається дійсним в рамках даного протоколу, тобто,

$$hash(hash(B_{prev}, A, t)) \leq bal(A)\frac{M}{D},$$

де

– $D \in [1,\ M]$ – складність даного блоку,

– A – адреса користувача,

– t – мітка часу,

– B_{prev} – попередній блок в ланцюгу.

Введемо значення $T(r)$ як кількість часу в секундах, необхідного для генерації блоку B мінтером, r – кількість спроб підбору значень в секунду. Згідно з умовами, які накладаються даним протоколом на мітку часу, допускається 7200 різних значень t, тобто, область перебору,

порівняно з PoW, значно менша. Зважаючи на це, будемо припускати, що кількість спроб $r = 1$.

Аналогічно попереднім обчисленням, будемо припускати, що всі дійсні блоки в протоколі консенсусу задовольняють умові:

$$U \leq \theta \leq 1,$$

де $U \sim Un[0,1]$ – рівномірно розподілена випадкова величина, отримана шляхом гешування певних даних та нормалізації даної величини таким чином, щоб її значення знаходилось на проміжку $[0,1]$. Proof of Stake – частинний випадок даної умови зі значенням $\theta = \frac{bal(A)}{D}$. Нехай N – кількість спроб, необхідних для отримання значення, що задовольняє нерівності. В такому випадку, можемо покласти $T(r) = T$.

Розглянемо розподіл величини T:

$\Pr\{\mathrm{T} \leq t\} = Pr\{N \leq t\} = 1 - Pr\{N > t\} = 1 - \left(1 - \frac{bal(A)}{D}\right)^t =$
$= 1 - \exp\left(ln\left(1 - \frac{bal(A)}{D}\right)t\right).$

Введемо додаткове припущення, згідно з яким баланс кожного мінтера є значно меншим, ніж загальна сума валюти в системі. В такому випадку, зважаючи на значення $D = \frac{1}{T_{ex}} \sum_a bal(a)$, можемо допустити, що $\frac{bal(A)}{D} << 1$, тому $ln\left(1 - \frac{bal(A)}{D}\right) \approx -\frac{bal(A)}{D}$. Тоді

$$Pr\{T \leq t\} = 1 - exp\left(-\frac{bal(A)}{D}t\right).$$

Нехай в криптовалютній системі є n учасників. Тоді час знаходження дійсного блоку B становить $T_b = min(T_1,...,T_n)$, де T_i – час, необхідний для отримання значення, що задовольняє умові, i-м мінтером. Тоді:

$$Pr\{T_b \leq t\} = 1 - exp\left(-\frac{t}{D}\sum_{i=1}^{n} bal(A_i)\right).$$

Отже, бачимо, що час, необхідний для отримання нового дійсного блоку розподілений експоненціально з параметром $\lambda = -\frac{\sum_{i=1}^{n} bal(A_i)}{D}$. Тоді за властивістю експоненціального розподілу, отримуємо імовірність

згенерувати дійсний блок мінтером i:

$$Pr\{T_b = T_i\} = \frac{bal(A_i)}{\sum\limits_{j=1}^{n} bal(A_j)}.$$

Це означає, що мінтінг у протоколі Proof of Stake є справедливим, тобто, мінтер з часткою балансу p має імовірність p згенерувати наступний дійсний блок.

3.3 Proof of Importance

Нехай B – запропонований новий блок, який вважається дійсним в рамках даного протоколу, що означає виконання наступної рівності:

$$2^{54}\left|ln\Big(\frac{hash(A, B_{prev})}{2^{256}}\Big)\right| < 2^{64}\frac{c \cdot imp_A}{D}t,$$

де

– $D \in [10^{13},\ 10^{15}]$ – складність даного блоку,

– A – відкритий ключ користувача,

– t – мітка часу,

– B_{prev} – generation hash попереднього блоку в ланцюгу,

– imp_A – рейтинг важливості облікового запису,

– $c = 8999999999$ – константа.

Зважаючи на те, що на конкретному періоді користувач може змінювати лише час і його значення $t > 0$, перепишемо нерівність як:

$$\frac{2^{54}}{t}\left|ln\Big(\frac{hash(A, B_{prev})}{2^{256}}\Big)\right| < 2^{64}\frac{c \cdot imp_A}{D}.$$

В такому разі, можемо припустити, що всі дійсні блоки в протоколі консенсусу задовольняють умові:

$$U \leq \theta \leq 1,$$

де $U \sim Un[0,1]$ – рівномірно розподілена випадкова величина, отримана шляхом гешування даних про відкритий ключ користувача і значення generation hash попереднього блоку та нормалізації даної величини таким чином, щоб її значення знаходилось на проміжку $[0,1]$. Proof of Importance – частинний випадок даної умови зі значенням $\theta = \frac{imp_A}{D}$.

Аналогічно попереднім дослідженням, введемо наступні позначення: $T(r)$ – час в секундах, необхідний для отримання пари значень, задовольняє нерівності $(hit < target)$, N – кількість спроб, необхідних для отримання значення, що задовольняє нерівності. Оскільки величина hit змінюється в залежності від часу, який пройшов з моменту приєднання попереднього блоку, в даному випадку також доцільно покласти $r = 1$, а $T(r) = T$. В такому разі маємо розподіл часу, необхідного для утворення нового блоку (обчислення аналогічні попереднім):

$$Pr\{T \leq t\} = 1 - exp\Big(-\frac{imp_A}{D}t\Big).$$

Таким чином, час, необхідний для створення нового блоку, розподілений експоненціально з параметром $\lambda = -\frac{\sum\limits_{i=1}^{n} imp_{A_i})}{D}$. При наявності в системі n учасників з правом приєднання блоку, отримає цю можливість той, хто за найменший час $T_b = min(T_1,...,T_n)$ згенерує пару значень таку, що $(hit < target)$. Відповідно, імовірність того, що i-й учасник згенерує її становить:

$$Pr\{T_b = T_i\} = \frac{imp_{A_i}}{\sum\limits_{j=1}^{n} imp_{A_j}}.$$

Отже, і в протоколі Proof of Importance харвестинг є справедливим, оскільки імовірність додавання нового блоку напряму залежить від

рейтингу важливості користувача.

Тепер розглянемо дану оцінку з точки зору користувача, який, використовуючи модифікований протокол та маючи достатню кількість пристроїв з унікальними IP адресами, виконує спробу приєднати наступний блок. В такому разі пара значень $(hit, target)$ повинна задовольняти наступній нерівності:

$$2^{54}\left|ln\left(\frac{hash(A, B_{prev})}{2^{256}}\right)\right| < 2^{64}\frac{c \cdot imp_A \cdot ln(m)}{D}t.$$

Тоді дана модифікація є частинним випадком умови, описаної вище зі значенням $\theta = \frac{imp_A \cdot ||ln(m)||}{D}$. Імовірність того, що учасник, який спромігся відновити значення секрету, описується наступною рівністю:

$$Pr\{T_b = T_i\} = \frac{imp_{A_i} \cdot ||ln(m)||}{\sum\limits_{j=1}^{n} imp_{A_j}},$$

де $||ln(m)||$ – нормоване у відповідності з припущенням значення логарифму. Отже, бачимо, що дана модифікація дійсно збільшує імовірність окремого учасника отримати дійсне значення нового блоку.

Висновки до розділу 3

В даному розділі було проведено порівняльний аналіз існуючих криптографічних протоколів досягнення узгодження та запропонованої модифікації, в результаті чого були отримані розподіли імовірностей створення нового дійсного блоку учасниками в кожному окремому випадку, які наведено нижче:

– Proof of Work – $Pr\{T_b = T_i\} = \frac{r_i}{\sum\limits_{j=1}^{n} r_j}$,

– Proof of Stake – $Pr\{T_b = T_i\} = \frac{bal(A_i)}{\sum\limits_{j=1}^{n} bal(A_j)}$,

– Proof of Importance – $Pr\{T_b = T_i\} = \frac{imp_{A_i}}{\sum\limits_{j=1}^{n} imp_{A_j}}$,

– Proof of Importance з використанням IP адрес – $Pr\{T_b = T_i\} = \frac{imp_{A_i} \cdot ||ln(m)||}{\sum\limits_{j=1}^{n} imp_{A_j}}$.

Згідно з даними результатами маємо наступні висновки:

– протоколи Proof of Work, Proof of Stake та Proof of Importance є справедливими з точки зору залежності успіху генерації дійсного блоку від кількості цінного ресурсу, наявного у користувача;

– у протоколах Proof of Work та Proof of Stake дана імовірність дійсно залежить лише від наявності цінного ресурсу та жодним чином не може бути змінена без привнесення ресурсу ззовні системи;

– Proof of Importance з використанням IP адрес дійсно збільшує імовірність учасника отримати право приєднати новий блок.

ВИСНОВКИ

В даній роботі було проведено огляд опублікованих джерел за тематикою дослідження, серед яких:

– опис принципів роботи технології «блокчейн»,

– специфікації протоколів досягнення узгодження в розподілених криптовалютних системах,

– схеми криптографічного розділення секрету з підтвердженням та без.

В результаті цього було виокремлено головні недоліки існуючих протоколів консенсусу та вибрано схему розподілу секрету, яка виявилась найбільш задовільною для побудови модифікацій.

Далі було обрано два напрямки, за якими виконувалась побудова модифікацій протоколу. Перший напрямок передбачає привнесення до протоколу консенсусу принципу лотереї. Це частково вирішує проблему того, що учасник без наявності певного цінного ресурсу має можливість періодично отримувати право на приєднання до ланцюга нового дійсного блоку. Другий напрямок націлений на учасників з правом генерації блоку та орієнтований на перерозподіл даного права протягом деяких періодів з метою уникнення зосередження його в межах певної групи.

В третій частині даної роботи було проведено дослідження існуючих протоколів досягнення узгодження, в результаті чого було наведено теоретичні оцінки імовірностей генерації та приєднання до ланцюга нового дійсного блоку, які описують залежність імовірності успіху кожного окремого учасника з правом генерації блоку в залежності від наявного в нього цінного для криптовалютної системи цінного ресурсу.

В майбутньому планується проводити дослідження в напрямку створення нових модифікацій криптографічних протоколів консенсусу, націлених на усунення проблем розподілу права генерації шляхом унеможливлення зосередження більшості цінного ресурсу у вузькому колі

облікових записів з допомогою використанням криптографічних схем розподілення секрету.

ПЕРЕЛІК ПОСИЛАНЬ

1. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. — Режим доступу: https://bitcoin.org/bitcoin.pdf.

2. Яковлєв С. В. Конспект лекцій з дисципліни «Спеціальні розділи криптографії». — 2017.

3. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. — 2015.

4. BitFury Group. Proof of Stake versus Proof of Work. Режим доступу: https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf

5. Iddo Bentov, Charles Lee, Alex Mizrahi, Meni Rosenfeld. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. — Режим доступу: https://eprint.iacr.org/2014/452.pdf

6. Proof of burn. — Режим доступу: https://en.bitcoin.it/wiki/Proof_of_burn

7. NEM Technical Reference. Version 1.2.1. — Режим доступу: https://nem.io/wp-content/themes/nem/files/NEM_techRef.pdf

8. Secret sharing. — Режим доступу: https://en.wikipedia.org/wiki/Secret_sharing

9. Брюс Шнайєр. Прикладна криптографія. 2 видання. — 2002

10. Ліфшиц Ю. Конспект лекцій з дисципліни «Сучасні задачі криптографії». — 2005.

11. Asmuth C., Bloom J. A modular approach to key safeguarding. — Режим доступу: https://ieeexplore.ieee.org/abstract/document/1056651

12. Иванцов А.М., Рацеев С.М. О применении эллиптических кривых в некоторых проверяемых схемах разделения секрета. — Режим доступу: http://apu.npomars.com/images/pdf/49_4.pdf

13. Torben Pedersen. Distributed provers with application to undeniable signature. — Режим доступу:

https://link.springer.com/content/pdf/10.1007/3-540-46416-6_20.pdf